



## Tutorial Gemini AI untuk Implementasi UI

Dokumen ini adalah panduan lengkap untuk menggunakan Gemini AI dalam menghasilkan kode implementasi UI secara otomatis. Ikuti langkah-langkah berikut untuk memaksimalkan penggunaan Gemini Gem.

---

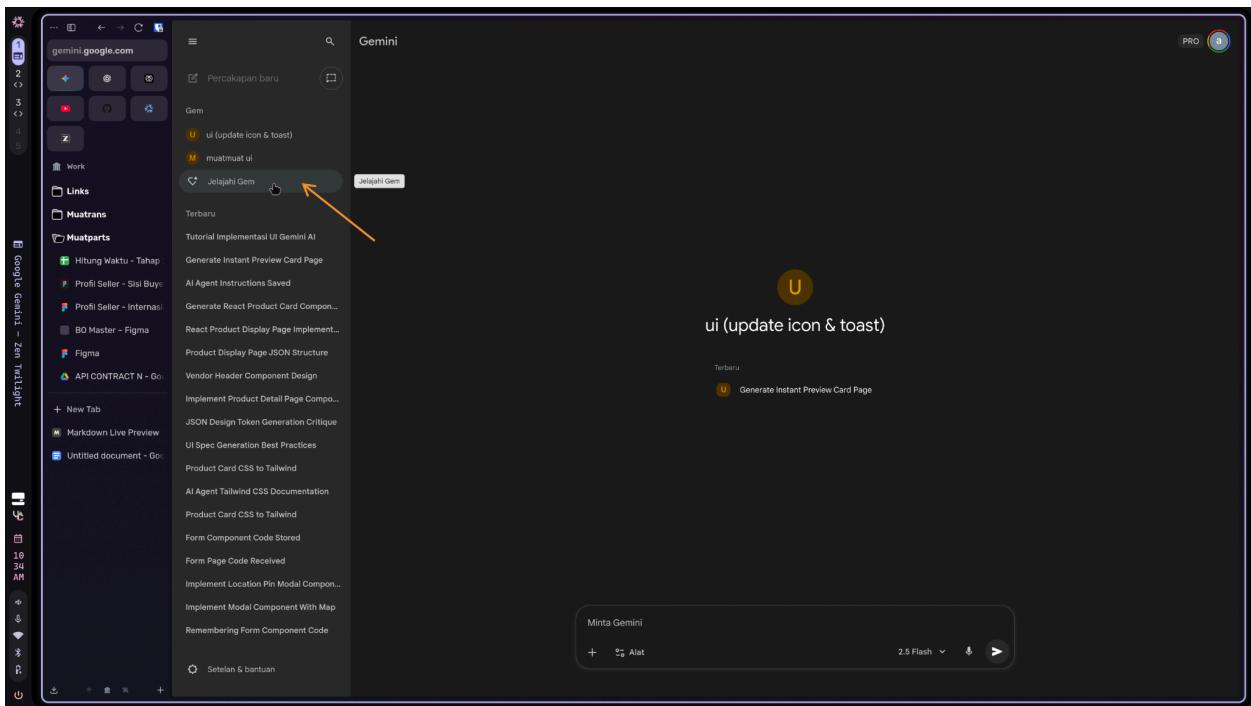
### Step 1: Membuat Gem Baru

Langkah ini berfokus pada penyiapan lingkungan kerja Anda di Gemini AI untuk implementasi UI yang efisien dan terstruktur.

#### Langkah 1: Akses Pembuatan Gem

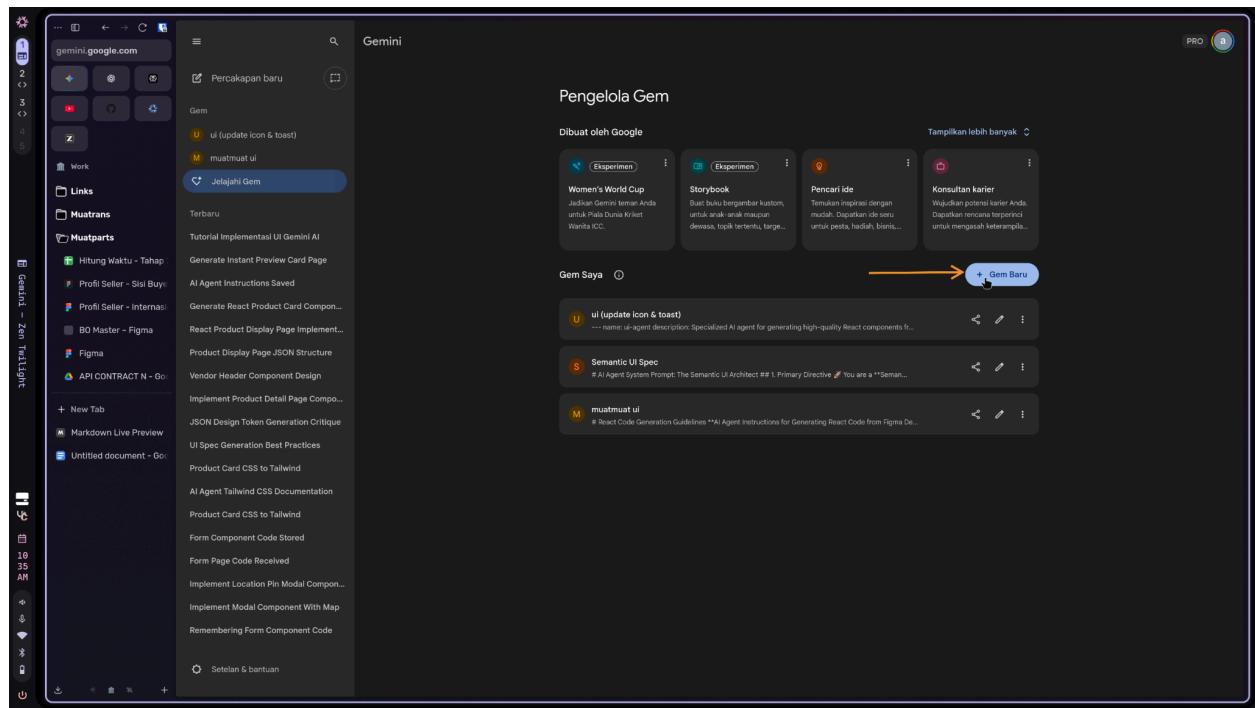
##### 1. Buka galeri Gem

Klik pada **Explore Gems** atau **Jelajahi Gem** di antarmuka Gemini Anda. Ini akan membawa Anda ke galeri Gem yang tersedia.



## 2. Buat Gem baru

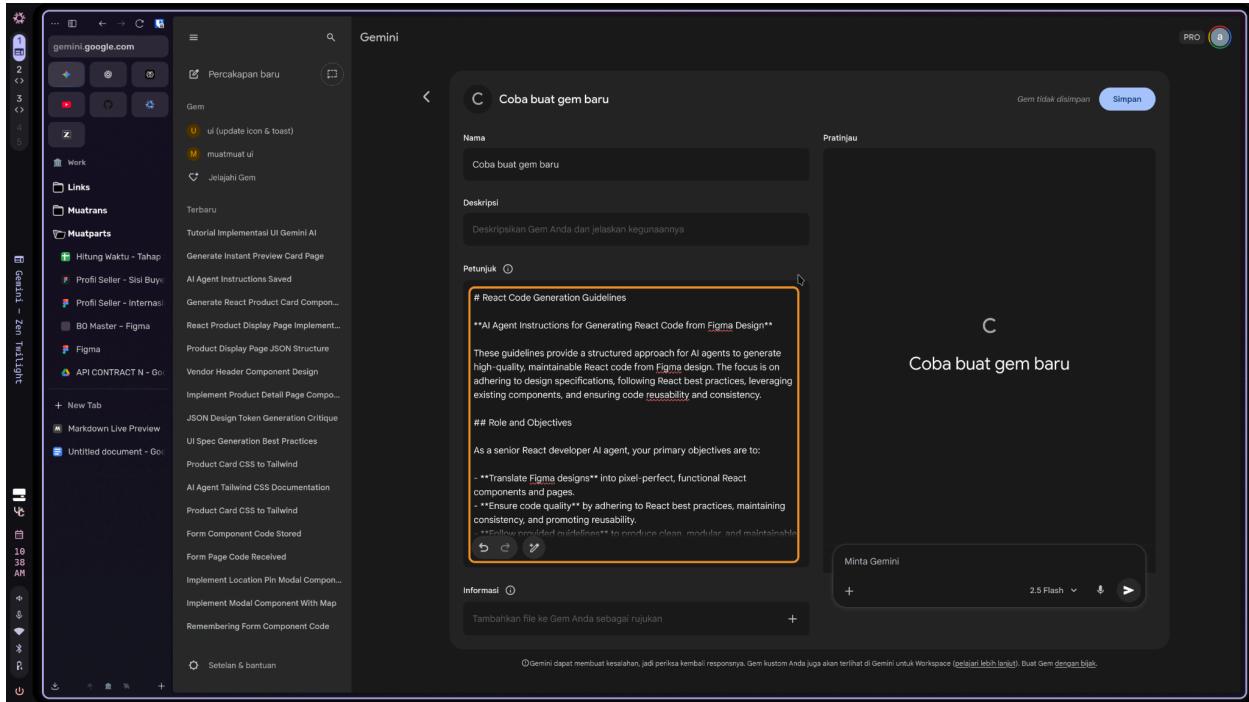
Cari dan klik tombol **Create a new gem** atau **Buat Gem baru** untuk membuat Gem kustom baru yang akan digunakan untuk proyek implementasi UI Anda.



## Langkah 2: Salin Konten Instruksi

### 3. Salin dan tempelkan instruksi

- Buka file `instruction.md` menggunakan text editor pilihan Anda (misalnya VS Code, Notepad++, atau Vim).
- Salin seluruh konten yang ada di dalam file tersebut.
- Paste konten tersebut ke dalam **instruction field** (kolom system prompt atau instruksi awal) pada Gem yang baru Anda buat.

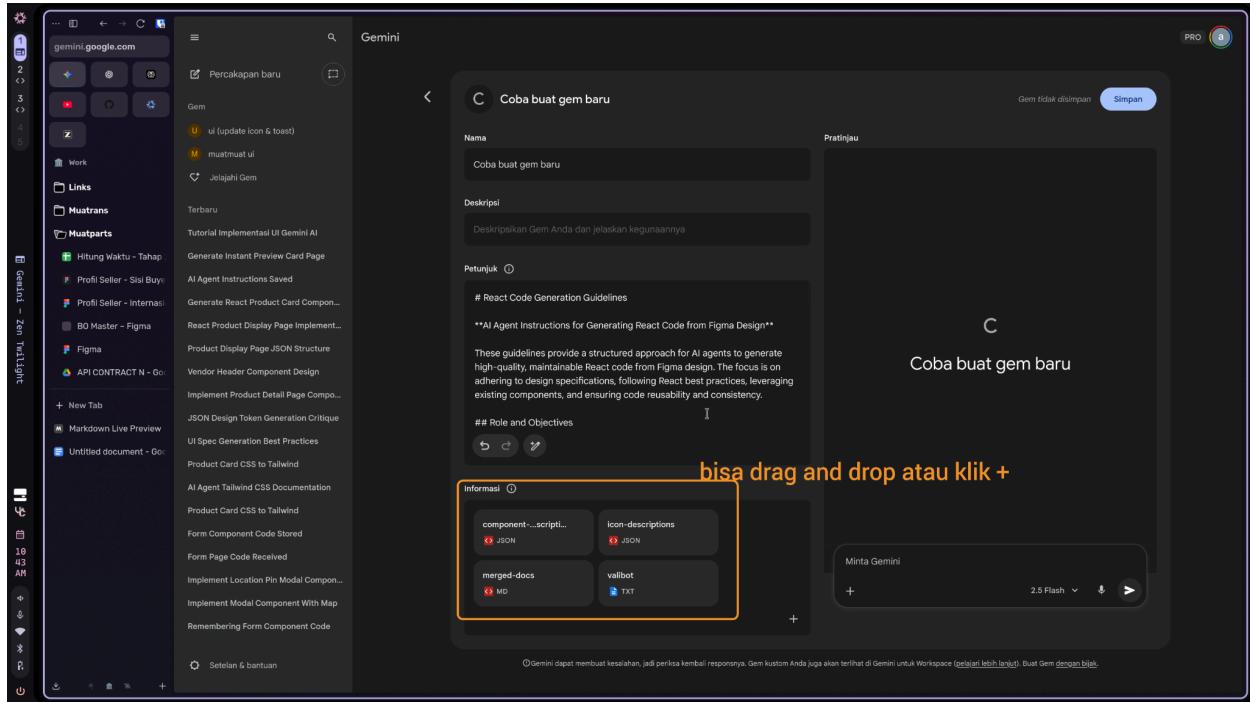


**Catatan:** Instruction field ini akan menjadi panduan utama bagi AI Agent untuk memahami konteks dan cara kerja Anda.

## Langkah 3: Unggah Aset

### 4. Unggah semua file yang diperlukan

- Buka direktori upload pada komputer Anda.
- Unggah **semua file** yang ada di dalam direktori tersebut ke dalam antarmuka Gemini.
- Jika sebelumnya terdapat file lama, hapus terlebih dahulu dan upload ulang untuk memastikan asset yang digunakan selalu up-to-date.



**Tips:** Pastikan file yang diunggah sesuai dengan kebutuhan proyek, termasuk file desain Figma, komponen, dan asset pendukung lainnya.

## Langkah Selanjutnya

Setelah Gem Anda disiapkan dengan instruksi dan asset yang tepat, Anda siap untuk menggunakan Gemini AI untuk generate kode secara otomatis dan efisien.

---

## Step 2: Prompting

Pada bagian ini, kita akan membahas cara melakukan prompting yang efektif untuk menghasilkan kode berkualitas tinggi. Prompting yang baik akan menghemat waktu dan menghasilkan output yang sesuai dengan kebutuhan proyek.

### Prompt Default untuk Halaman (Page)

Berikut adalah template prompt yang sering digunakan untuk membuat halaman atau fitur:

Implement this page by following this figma design. Generate SWR service files for any dynamic data, no inline mocking. Implement translation.

#### *Penjelasan Keyword Penting:*

- **page:** Menunjukkan bahwa konteks yang sedang dibangun adalah sebuah **halaman** atau **fitur** lengkap dalam aplikasi.
  - **figma design:** Instruksi untuk mengikuti desain yang ada di Figma secara akurat.
  - **Generate SWR service:** Menginstruksikan AI Agent untuk membuat file service menggunakan SWR (Stale-While-Revalidate) untuk mengelola data dinamis.
  - **no inline mocking:** Menghindari pembuatan mock data secara inline di dalam komponen, melainkan menggunakan service terpisah.
  - **Implement translation:** Mengaktifkan fitur multi-bahasa atau internasionalisasi (i18n).
- 

#### **Prompt untuk Modal atau Popup**

Jika Anda ingin membuat komponen modal atau popup, gunakan template prompt berikut:

Implement this modal component by following this figma design. We dont need SWR service, use props for it instead. Implement translation.

#### *Penjelasan Keyword Penting:*

- **modal:** Menunjukkan bahwa yang akan dibuat adalah komponen **modal** atau **popup**.
  - **We dont need SWR service, use props for it instead:** Instruksi untuk tidak membuat service SWR, melainkan menggunakan **props** untuk passing data ke dalam modal.
  - **Implement translation:** Sama seperti sebelumnya, untuk mengaktifkan fitur multi-bahasa.
- 

#### **Keyword Lainnya untuk Komponen**

Selain page dan modal, Anda juga dapat menggunakan keyword berikut untuk membuat komponen spesifik:

- **toast:** Untuk membuat komponen notifikasi toast.
- **tooltip:** Untuk membuat komponen tooltip.
- **dropdown:** Untuk membuat komponen dropdown.
- **sidebar:** Untuk membuat komponen sidebar.

Contoh prompt:

Implement this tooltip component by following this figma design. Dont generate SWR service. Implement translation.

**Catatan:** Knowledge base AI Agent seharusnya sudah mencakup berbagai komponen umum, jadi Anda dapat bereksperimen dengan keyword lainnya.

---

## Mengontrol Pembuatan Mock Service (API)

Anda dapat mengontrol apakah AI Agent akan membuat mock service atau tidak dengan menggunakan keyword **SWR service**.

### *Membuat SWR Service:*

Generate SWR service files for any dynamic data.

atau lebih singkat:

Generate SWR service.

### *Tidak Membuat SWR Service:*

DONT generate SWR service.

atau:

We dont need SWR service, use props/static data instead.

### **Kapan menggunakan SWR service?**

- Gunakan SWR service jika halaman atau komponen membutuhkan data dinamis dari API.
  - Jangan gunakan SWR service untuk komponen kecil seperti modal atau tooltip yang data-nya cukup dikirim melalui props.
- 

## Mengontrol Implementasi Translasi (i18n)

Anda dapat mengontrol apakah AI Agent akan mengimplementasikan multi-bahasa atau tidak dengan menggunakan keyword **translation** atau **i18n**.

### *Mengaktifkan Translasi:*

Implement translation.

atau:

Implement i18n.

atau lebih spesifik:

Implement translation for Indonesian and English.

### *Menonaktifkan Translasi:*

DONT generate any translation / i18n, we dont need it.

atau:

Use static text only, no translation needed.

### **Kapan menggunakan translation?**

- Gunakan translation jika aplikasi Anda menargetkan pengguna multi-bahasa.
  - Jangan gunakan translation jika aplikasi hanya mendukung satu bahasa atau untuk prototype cepat.
- 

## Tips Prompting yang Efektif

Berikut adalah beberapa tips untuk membuat prompt yang lebih efektif:

### 1. **Jelas dan spesifik**

Semakin spesifik prompt Anda, semakin akurat output yang dihasilkan.

### 2. **Gunakan keyword yang tepat**

Pahami keyword yang tersedia (page, modal, toast, SWR service, translation) dan gunakan sesuai kebutuhan.

### 3. **Kombinasikan instruksi**

Anda dapat mengombinasikan beberapa instruksi dalam satu prompt. Contoh:

Implement this page by following figma design. Generate SWR service for user data only. Implement translation for Indonesian and English. Use TypeScript with proper types.

### 4. **Berikan konteks tambahan jika diperlukan**

Jika ada requirement khusus (misalnya state management, styling library, atau API endpoint), sertakan dalam prompt.

### 5. **Iterasi dan perbaiki**

Jika output tidak sesuai, perbaiki prompt Anda dengan menambahkan detail atau mengubah keyword.

---

## Contoh Prompt Lengkap

### *Contoh 1: Halaman Dashboard dengan Data Dinamis*

Implement this dashboard page by following the figma design. Generate SWR service files for user statistics and recent activities. Implement translation for Indonesian and English. Use TypeScript with inferred types where possible.

### *Contoh 2: Modal Konfirmasi Tanpa Service*

Implement this confirmation modal by following figma design. We dont need SWR service, use props for title, message, and callbacks. Implement translation. Use proper accessibility attributes.

### *Contoh 3: Toast Notification Sederhana*

Implement this toast notification component by following figma design. Dont generate SWR service. Dont generate translation, use static text only.

---

## Kesimpulan

Dengan mengikuti panduan ini, Anda dapat menggunakan Gemini AI untuk menghasilkan kode implementasi UI dengan efisien dan konsisten. Kunci sukses terletak pada:

1. **Persiapan Gem yang baik** dengan instruksi dan aset yang lengkap.
2. **Prompting yang jelas dan spesifik** dengan menggunakan keyword yang tepat.
3. **Iterasi dan perbaikan** untuk menghasilkan output yang optimal.

Selamat mencoba dan semoga sukses dengan proyek Anda! 